A REVIEW ON HADOOP

Dr Sanjay Tyagi[1], Monika Kumari[2]

[1] Department of Computer Science and Applications, Kurukshetra University, Kurukshetra,Haryana,India
tyagikuk@yahoo.com
[2] Department of Computer Science and Applications, Kurukshetra University, Kurukshetra,Haryana,India
monikakeshu@gmail.com

***Abstract*— The emergence of new data sources and the need to consider everything, even the unstructured data has led many organisations to a deduction that a single data warehousing scheme now cannot handle the growing depth of analytical workloads. Incoming data volumes are exploding in complexity, variety, volume and speed. Being purpose-built for big data analytics, Hadoop is now becoming a must addition to data warehouse environment, where it fulfils several roles. With this, business would be able to understand data quickly and also to explore this data for value, allowing analyst to ask and reprise their business questions quickly. Hadoop, built to allow certain forms of batch oriented distributed data processing, lends itself readily to the assimilation process. But, it was built on fundamentals which severely limit its ability to act as an analytic database.**

***Index Terms*— Data deluge, Hadoop, HDFS, MapReduce, RDBMS**

I. INTRODUCTION

Every industry across the globe has to face same challenge i.e. their data arrives faster than the present data warehousing and they have to capture it and then analyze it rapidly. Unexpected volumes of click streams data and transaction is driven from migration to online channels. They are run to drive up the data warehousing cost, analytics, processing and ELT (Extract, Transform and Load).

The major challenge here is unstructured data. Most of the businesses now want to analyze very complex and high valued data types like social media data, multi structured data, un-modelled data and clickstreams for earning new insights. But the main issue is that these data types never fit the existing parallel processing models which are designed for the structured data for major data warehouses [1]. The price in scaling traditional data warehousing technologies is very elevated and ultimately it becomes undesired. But, when the cost can be justified easily, then the performance has to insufficiently adapt present velocity, variety of data and growing volume. Two major technologies are needed, i.e. cost effectiveness and scalability. Only Hadoop satisfies both these needs.

In a cluster of commodity computers, Hadoop is one of the complete open source ecosystems for processing disparate data sources, visualizing, analyzing, sharing, searching, sorting and organizing. A virtually unlimited scalability and availability is provided by this architecture. It is served by few thousand servers and each of them offers computation and local storage.

Hadoop has the ability to analyze and store large set of data in parallel. All this is done on large clusters of computers, which yields exceptional performance, but the use of commodity hardware always results in remarkable low cost. The cluster of Hadoop often price from 50 to 100 times lower on per terabyte basis than any data warehouse system. Along with the entire performance and price ratio, there is not any surprise that Hadoop is changing the paradigm of data warehousing [2].

A. Hadoop background

A collection of open source projects is known as Hadoop. Basically, it was originated by Dough Cutting in 2006 for applying the framework of Google MapReduce programming across the distributed system. At its core, there are two components, the first is the MapReduce (a programming and job management framework) and the second one is the Hadoop Distributed File System (HDFS). All this is because the Hadoop provides an easy obtainable framework for the purpose of distributed processing. Many numbers of open source projects are quickly emerging, which are leveraging this to solve many specific problems [3].

Products of Hadoop include Zookeeper, Impala, Chukwa, Avro, Pig, Ambari, YARN, Cassandra, Mahout, Hbase, Hive, MapReduce and Hadoop Distributed File System (HDFS). But note that the MapReduce products never require HDFS and most of them run atop with relational DBMS [4]. A general purpose execution engine handles the complexity of parallel programming for a large range of hand code logics and many other applications that includes analytics [5].

An open source framework for developing distributed applications is Apache Hadoop. It has the capability to process a very huge amount of data. This platform provides both computational capabilities and distributed storage. Two main working layers of Hadoop are:-

1) Distributed storage layer: It has a distributed file system known as HDFS which provides storage.
2) Computational layer: An important framework called MapReduce is used by this layer.

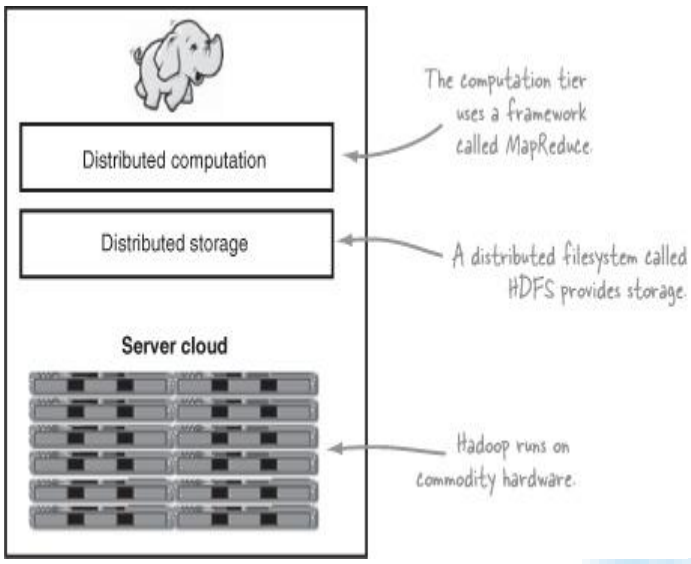

Figure1. Hadoop layers. (Guruzon, 2014)

The key features of distributed computing, which have proved Hadoop very desirable and distinctive, are:

1. Accessible: Working of Hadoop is done on cloud computing services so data can be accessed on various nodes.
2. Robust: The designing of Hadoop is done to run on commodity hardware. The architecture of the same is done with the assumption of frequent hardware malfunctions. Most of such failures are smoothly handled by it [6].
3. Scalable: By adding more nodes to the cluster, Hadoop scales more linearly for handling the data.
4. Simple: It allows users to write efficient parallel code immediately. The simplicity and accessibility of Hadoop gives it an edge for running and overwriting large distributed programs [7].

B. Role of Hadoop in New Data Warehousing Paradigm

1. Data staging: Role of Hadoop in data warehousing is emerging very rapidly. Firstly, it was used as a platform for loading, transforming and extracting data. In this, Hadoop is often used to offload transformations and processing in the data warehouse. So it replaces ELT i.e. Extract, Load and Transforms, which needs loading data in the data warehouse as a method to perform large scale and complex transformations. Along with Hadoop, the data is to be extracted and loaded in the Hadoop cluster, where it can be easily transformed in real time along with the loaded results in the data warehouse for future analysis. In all sprites the process of ELT has evolved as a method of taking advantage of the parallel query processing situated on the platform of the data warehouse.
2. Data archiving: In "front end", Hadoop has a major role to play. Basically, it works in performing transformation processing. In "back end", it works in offloading data from the data warehouse. With the help of virtual scalability at per terabyte rate, this is 50 times higher than the traditional data warehouse. Hadoop also functions well for data archiving and this is only because Hadoop can operate on analytics of the archived data. It is also mandatory to move on the specific result set for data warehouse's analysis of future [2]. Generally, the enterprise has three options, when it comes to archiving the data. It has to be left with in the relational database or move it to the tape or it can be deleted. The scalability of Hadoop and with its low cost enabled function, organizations can keep all data forever in a readily manageable online environment.
3. Schema flexibility: For stable semi structured data (JSON and XML) and highly structured data (CRM and ERP), relational DBMS are well equipped. As, complement Hadoop can easily and quickly ingest any kind of data format including the no schema (images, video and audio) and evolving schema (like in A/B and multivariate tests on a website).
4. Processing flexibility: Nosql approach of Hadoop is a more natural framework for the manipulation of non traditional data types. It also enables procedural processing which is valuable in use case like gap recognition and time series analysis. Hadoop also supports lots of programming languages, which provide a huge capability than SQL. Along with it, Hadoop also enables the increasing practice of "late binding" apart from transforming the data as ingested by it. The structure of the same is applied in runtime [4].

II. *HADOOP AND RDBMS*

At very high level we can say that SQL (structured query language) is by design targeted at structured data only but most of initial applications in Hadoop designed to deal with unstructured data.

A detailed comparison of Hadoop with SQL databases under specific dimensions is as under:

1. Scale-out instead of scale-up: Scaling a organisation's relational databases is expensive because to maintain a bigger database you need bigger system. Hadoop is drafted as a scale-out architecture operating on a cluster of commodity hardware, adding more resources means adding more machines to the cluster. A cluster with ten to hundreds of machines is standard.
2. Key/value pair database design instead of relational tables: In RDBMS, data is stored in tables having relational structure defined by a schema. Hadoop uses key/value pairs as its basic data unit, which works well with the less-structured data type. In Hadoop, data can originate in any form, but it eventually transforms into (key/value) pairs for the processing functions to work on [6].
3. Functional programming (MapReduce) instead of declarative queries (SQL): **SQL has query statements; while under MapReduce scripts and codes**. MapReduce allows processing data in a more general fashion than SQL queries. For example, Hadoop can build complex statistical models from

data or reformat the image data. SQL cannot handle such tasks.

4. Offline batch processing instead of online transactions: Hadoop is **designed for offline processing and analysing large-scale data**. It does not perform random reading and writing of a few records, which is the type of load in online transaction processing. Hadoop is best used as a write-once, read-many-times type of data storage. In this way it is similar to data warehouse design in the SQL world [7].

III. POSITIVE AND NEGATIVE CONSEQUENCES

Following are the list of areas where Hadoop is found very stable:

1. Hadoop provides Distributed storage & Computational capabilities together [8].
2. Hadoop is a highly scalable storage platform, it stores and distributes very large data sets across hundreds of nominal servers that operating in parallel. Unlike (RDBMS) that can't scale to process large amounts of data [9].
3. Alternative high performance computing (HPC) systems allow programs to run on large collections of computers, but they usually require strong programming configuration and that data be stored on a separate storage area network system. Careful administration is required for Schedulers on HPC clusters and since program execution is sensitive to node failure, administrating a Hadoop cluster is much easier [10].
4. HDFS uses large size blocks that ultimately help scalability. It works best when manipulating large files (gigabytes, petabytes...).
5. Scalability and Availability are the specified features of HDFS to achieve data replication and fault tolerance system.
6. HDFS can replicate files for specified number of times (default is 3 replicas) so that it can tolerate software and hardware failure, moreover, it can automatically recover data blocks on nodes that have failed [11].
7. Hadoop uses MapReduce, a batch-oriented, distributed computing framework, which allows parallel work over a large amount of data.
8. With MapReduce, the developers focus only on addressing business needs, instead of getting involved in distributed system complications.
9. For parallel & faster execution of the Jobs, MapReduce splits the job into Map & Reduce tasks and schedules them for remote execution on the slave nodes or data nodes of the Cluster [7].

Following are the common areas found as soft spot in Hadoop framework:

1. Hadoop uses HDFS and MapReduce, and both these master processes have single point of failure, though work going on for high availability versions like multi server model [12].
2. Security is also one of the major concerns, because Hadoop does offer a security model, but by default it is disabled because of its high complexity. Once authenticated to a hadoop cluster, a user has all the data in that cluster [13].
3. Storage encryption and network level encryption are not offered by Hadoop which is very big concern for government sector application data [14].
4. HDFS is inefficient for handling small files. It lacks transparent compression as HDFS is not planned to work well with random reads over small files due to its optimization for prolonged throughput.
5. MapReduce is a shared-nothing architecture, hence tasks that require global synchronization or sharing of mutable data, are not a good fit which can pose challenges for some algorithms [7].
6. **Backup** is also difficult. Hadoop is fault tolerant, but enterprises still want a disaster recovery plan, or to go back to a point in time back up where some human error should result in data corruption.
7. **Real-time query** is not an attribute of Hadoop. While an emerging set of SQL-based languages and caching layers have been created, Hadoop is still inappropriate for real time computing [8].

None of these issues limit Hadoop, but failure to acknowledge these limitations may lead to unrealistic expectations that cannot be fulfilled with Hadoop.

IV. CONCLUSION

The data flood with its three equally challenging dimensions variety, volume, and velocity has challenged a single platform to meet all the organization's data warehousing requirements. Hadoop does not replace relational databases, but it's much better price/performance ratio will give organizations an opportunity to lower costs while maintaining their existing applications and reporting infrastructure.

So get started with Hadoop at the front end i.e. extract, transform and load, at the back end with an Active Archive, or in-between by integrating existing technologies with Hadoop's parallel processing prowess for both structured and unstructured data depending on your greatest need. Those who are still reluctant to make investment at this time, consider getting started in the cloud, where Hadoop is now available as an "on-demand" service.

So, as soon as your organization starts, be prepared to become a believer in the new multi-platform data warehousing paradigm, and in Hadoop as a potential and powerful enterprise data management hub.

REFERENCES

[1] J. L. A. A. J. T. Stefano Rizzi, "Research in data warehouse modelling and design: dead or alive?", *ACM,* Vols. 1-59593-530-4, no. 06, p. 0011, 2006.
[2] J. Norris, "How (and Why) Hadoop is Changing the Data Warehousing Paradigm", TDWI, 13 August 2013. [Online]. Available:http://tdwi.org/articles/2013/08/13/hadoop-changing-dw-paradigm.aspx. [Accessed January 2014].
[3] "Hadoop's Limitation for Big Data Analytics", ParAccel, 2012.
[4] P. Russom, "Where Hadoop Fits in Your Data Warehouse Architecture", TDWI, 2013.

[5] P. Russom, "Integrating Hadoop into Business Intelligence and Data Warehousing", TDWI, 2013.
[6] C. Lam, "Hadoop in Action", Manning, 2010, pp. 7-8.
[7] "Hadoop Introduction", Guruzon, 2014. [Online]. Available: http://guruzon.com/6/introduction/hadoop/what-is-hadoop-apache-cluster-bigdata-use-limitation. [Accessed February 2014].
[8] G. Harrison, "Why Hadoop project fail- and how to make yours a success", venturebeat, 24 June 2013. [Online]. Available: http://venturebeat.com/2013/06/24/why-hadoop-project-fail-and-how-to-make-yours-a-success. [Accessed February 2014].
[9] M. Nemschoff, "Big Data: 5 major advantages of Hadoop", ITProPortal, 20 December 2013. [Online]. Available: http://www.itproportal.com/2013/12/20/big-data-5-major-advantages-of-hadoop/. [Accessed February 2014].
[10] P. Conrad, "Apache Hadoop | MapR", MapR, 2013. [Online]. Available: http://www.mapr.com/products/apache-hadoop. [Accessed 20 February 2014].
[11] "Hadoop advantanges and disadvantages", java J2EE Tutorials, [Online]. Available: http://www.j2eebrain.com/java-J2ee-hadoop-advantages-and-disadvantages.html. [Accessed February 2014].
[12] S. Radia, "Apache Hadoop-High Availability", Apache Software, [Online]. Available: http://hadoop.apache.org/docs/r2.3.0/hadoop-yarn/hadoop-yarn-site/HDFSHighAvailabilityWithNFS.html. [Accessed March 2014].
[13] Jason C. Cohen,Dr. Subrata Acharya, "Incorporating Hardware Trust Mechanism in Apache Hadoop", *IEEE*, Vols. 978-1-4673-4941, no. 3, p. 12, 2012.
[14] Hsiao-ying Lin, Shiuan-Tzuo Shen, Wen –Guey Tzeng,Bao-Shuh P.Lin, "Towards Data Confidentiality via Integrating Hybrid Encryption Schemes and Hadoop Distributed File System", *IEEE*, Vols. 1550-445, no. X, p. 12, 2012.